THEORETICAL ASPECTS OF LEARNING IN NEURAL COMPUTATION

Călin ENĂCHESCU

Abstract

Neural networks are appropriate mathematical tools for the prediction of the nonlinear components of time series. We presume that there are functional relations among past, present and future values of time series. We have analyzed the behavior of neural networks in the prediction of different economical time series, building a special architecture of neural networks.

Key-words: neural networks, economical and financial time series, prediction, approximation.

1. INTRODUCTION

The economical land financial processes have major importance in the life and in the evolution of the society. Their understanding is vital for the companies and other functional economic entities.

In many cases, the classical stochastic modeling techniques do not produce satisfactory results (i.e., many important events are not predicted by these). In this respect, we have to find other methods that can deal with inherent nonlinearities in treating these nonlinear economic forecasting problems.

The neural networks, known as tools for nonlinear approximation [4], regression and prediction are prime candidates to be used in the context of these new nonlinear approaches to economic and financial predictions.

1.1. Objectives

The main objective of the paper is to present some results about the artificial neural networks in the context of nonlinear economic and financial modeling and prediction. We aim to identify and present in a rigorous way the fundamental properties of the neural networks and to show the relevancy of these properties in the context of practical applications.

1.2. Problem definition

The paper is based on the assumption that the modeling and prediction of the economic and financial processes can be interpreted as the modeling and prediction of the time series that describe these processes.

The main idea is that the modeling and prediction of time series can be conceived as approximation problems and specifically as minimal approximation problems [2], where the "minimal" term refers to the complexity of the models. Consequently, the theoretical part of the thesis deals with the analysis of the approximation properties of the artificial neural networks and with the construction of neural network models with minimal complexity [3].

1.3. Approximation properties of artificial neural networks

We have analyzed the learning capabilities of neural networks [6], arguing that the task of learning from examples can be considered in many cases to be equivalent to multivariate function approximation, that is, to the problem of approximating a smooth function from sparse data, the examples (training set).

Our approach is based on the recognition that the ill-posed problem of function approximation from sparse data must be constrained by assuming an appropriate prior on the class of approximating functions.

In this context, we have to be more precise in establishing the meaning of "ill-posed problem of function approximation from sparse data" [8].

Definition 1: The problem of approximating a function f: X $\rightarrow$ Y is well-posed if the following conditions are satisfied:

(C1) Existence: For any x $\in$ X, there exists y $\in$ Y with $y = f(x)$;

(C2) Uniqueness: For any x, y $\in$ X, $f(x) = f(y)$ if and only if x = y;

(C3) Continuity: function f is continuous.

Definition 2: The problem of approximating a function f: X $\rightarrow$ Y is ill-posed if the one of the conditions (C1), (C2), (C3) is not satisfied.

From this point of view, the learning process of a neural network, or the approximation of a smooth function using a set of examples, called training set, is ill-posed. Usually the training set doesn't contain enough information; therefore the condition of uniqueness (C2) is not satisfied. In many cases, the training set contains noisy data that implies that the condition of continuity (C3) set is not satisfied.

Concluding, we can say that the learning process of a neural network is equivalent to the approximation of a smooth function from examples (the training set) [13].

1.4. Regularization neural networks

Regularization techniques typically impose smoothness constraints on the approximating set of functions. It can be argued that some form of smoothness is necessary to allow meaningful generalization in approximation type problems. A similar argument can also be used in the case of classification where smoothness involves the classification boundaries rather than the input-output mapping itself.

Our use of regularization, which follows the classical technique, introduced by Tikhonov [17], [18], identifies the approximating function as the minimizer of a cost functional that includes an error term $\frac{1}{2}\sum_i (z_i - y_i)^2$ and a smoothness functional $\frac{1}{2}\Phi[f]$, usually called a stabilizer.

Suppose that the training set $T = \{(x_i, z_i) | i = 1,2,\ldots,N\}$ has been obtained by random sampling of a function f, belonging to some space of functions X defined on Rn, in the presence of noise, and suppose we are interested in recovering the function f, or an estimate of it, from the set of data T. This problem is clearly ill-posed, since it has an infinite number of solutions. In order to choose one particular solution we need to have some a priori knowledge of the function that has to be reconstructed. The most common form of a priori knowledge consists in assuming that the function is smooth, in the sense that two similar inputs correspond to two similar outputs.

The main idea underlying regularization theory is that the solution of an ill-posed problem can be obtained from a variational principle, which contains both the data and prior smoothness information. Smoothness is taken into account by defining smoothness in such a way that lower values of the functional correspond to smoother functions.

Since we look for a function that is simultaneously close to the data and also smooth, it is natural to choose as a solution of the approximation problem the function that minimizes the following functional [7]:

$$H[f] = \frac{1}{2}\sum_i (y_i - z_i)^2 + \frac{1}{2}\lambda\Phi[f] \qquad (1)$$

λ is a positive number that is usually called the regularization parameter. The first term is enforcing closeness to the data, and the second smoothness, while the regularization parameter controls the trade off between these two terms. It can be shown that, for a wide class of functional (1), the solutions of the minimization of the functional (1) all have the same form [5].

The function that minimizes the functional (1) has the following form [15]:

$$f(\mathrm{x}) = \sum_i w_i G(\mathrm{x}; \mathrm{x}_i) + p(\mathrm{x}) \qquad (2)$$

where p(x) is a term belonging to the null space of the regularization term $\frac{1}{2}\Phi[f]$.

In practical application we can consider classes of stabilizers with void null space. Therefore, without reducing the generality of the solution of the minimization of functional (1), we can consider the following practical solution:

$$f(\mathrm{x}) = \sum_i w_i G(\mathrm{x}; \mathrm{x}_i) \qquad (3)$$

The solution (3) of the variational problem (1) has a simple interpretation in terms of a neural network with one layer of hidden units, of Multy Layer Perceptron (MLP) type [10].

2. DATA SERIES PREDICTION WITH NEURAL NETWORKS

2.1 General problems

Economic problems have an important role in the society life, activity of companies and people. To better understand these problems and to find the desired solutions, economists try to build models that describe the investigated problems. Using some developed models and predictions, studies are made to find the optimal solution.

Usually, data accessible to the economists are sets of numeric values that describe the situations about the investigated problem in a certain moment. Numeric data sets are called *data series*, and their analysis and values prediction is called the *analysis* respectively the *prediction of data series*. Examples of data series are: the value of monthly unemployment, the value of monthly / annual inflation, the value of different stocks and the value of daily exchange between different currencies. To develop those economic models we have to study the analysis of economic time series, and for the decisions process we have to make the prediction for those series, using the developed models.

An important class of economic data series is represented by financial data series. These series contain data that represent monetary values of some economic objects or reports on some monetary values of economic objects. In this way, these series play an important role in monetary decisions of the central bank, in the investments made by companies and private people, in stock exchange transactions.

2.2 The analysis and prediction of data series

A simple model for data analysis is based on the auto-regressive model of *order p*, noted by *AR(p)* and the average sliding model of *order q*, denoted by *MA(q)*. In the auto-regressive case the model is written [1]:

$$X_t = a_1 X_{t-1} + \dots + a_p X_{t-p} + Z_t \quad (4)$$

where *ai* are the coefficients of the model, and Z_t are the probabilistic variables that follow the same normal probabilistic distribution with 0 average and σ^2 variation.

The probabilistic average models have the following formula [1]:

$$X_t = b_0 Z_t + \dots + b_q Z_{t-q} \quad (5)$$

where *bi* are the parameters of the model, and Z_t is following the same normal probability rule $N(m, \sigma)$.

The combined model of *AR(p)* and *MA(q)* is the model *ARMA(p,q)* that contains both components, having the following formula:

$$X_t = a_1 X_{t-1} + \dots + a_p X_{t-p} + Z_t + b_0 Z_t + \dots + b_q Z_{t-q} \quad (6)$$

The prediction of data series is based on the supposition that there exist a functional relations between past, present and future data series values. Usually the supposition is that the functional relation is not completely deterministic, but also contains a stochastic component. In several cases, especially in the case of financial time series, it is assumed that the deterministic component is not dominant, the stochastic component being of great importance.

The idea of applying neural networks has its origins in the observation that functional relations between time series values are in most cases non-linear. Thus, for the approximation of these non-linear functional relations neural networks are very efficient [2], [20].

2.3 Predictions at microeconomic level

A large usage of predictions using financial data series is at microeconomic level. In this category we can consider predictions regarding exchange rates, predictions regarding different financial products, predictions regarding the value of products and commercial products (gold, aluminum, etc.). The usual users of these predictions are investors and brokers.

The problem of prediction at microeconomic level is the presence of the stochastic components. In this case one can observe relatively frequent shocks produced by more or less independent events, independent from the data series object (quick changes of values caused by wars and in general by political events). Another special phenomenon of these series is the presence of short or medium time trends that can disappear in short time.

2.4. Data processing

Data processing represents the process of data transformation before building the models. These transformations are conversions, classifications, filtrations or other similar processing.

Many times data are not properly structured to build predictive models. In these cases it is important to remove components that represent redundant or irrelevant information [16].

Another important aspect is the necessity to validate the proposed models. For this reason we need to have data to build the model and data to validate the model. Thus, it is necessary to divide data for training data and data for validation.

2.4.1. Processing methods

The general purpose of preprocessing is to remove the observable deterministic relations. Theoretically, the purpose is to obtain some data series with mean 0 and a small variation.

The first step in data preprocessing is to make comparable the components of the data series. For this purpose it is necessary to rescale the data so that the values of the data series components to have values in the interval [0,1] or [-1,1].

The second step is to remove the primary deterministic components that are easy to be observed. Examples of these types of components are trend and seasonality [12].

In the next step we can perform a data filtering. The purpose of the filtering process is to remove non-trivial periodical components that have dominant effects in data series. To determine those periodical components we can apply a Fourier transformation of the data series. To filter these components we can build linear filters. The most common filters are the low-pass filters, high-pass filters and band-pass filters. By combining those filters can be filtered outside non-trivial periodical components existing in data series [19]. These filters usually have the following form:

After removing the deterministic observable components we have the input data vectors that contain relevant information. On one side this is done by removing the components of data vectors that are not meaningful according to the known knowledge, on the other side data vectors are combined to obtain new vectors with more complete informational content. The question is, which are the previous data that influences the values of the data series in a future moment. This doesn't mean to search for functional relations but also to the search for relevant information. This problem is also called the problem of *determination of data dimension*.

Finally, input data vectors are analyzed to determine possible clustering [7]. This is done through simple classification of data. If we can group the data in precise distinct classes, separate models are built for those classes. It is possible, that after clustering the models to be equivalent. We must verify model equivalence, and in case of equivalent relations, we must build simplified data general models [12].

2.4.2. Data for training and validation

Analyzing the data corresponding to a problem it is possible to build predictive models. It is very important to test the accuracy of the generated model. Through validation we understand the testing of the model and the measurement of its performance using a measure of performance [14].

A method for splitting data in training data and in validation data is the simple division based on data feature. We can consider the data x_t with $t <= T_0$ as training data and data x_t with $t > T_0$ as validation data.

A practical observation is that the selection of the model depends on the validation set. A way to select the best model is to use more than one set of validation, for example the validation data are grouped in many validation sets. After that, the generated models are verified with every validation test and a balanced validation is generated. Theoretically, it is desired that the validity of the models to be tested for every possible validation set. But this is not possible practically, because of the necessity of excessive calculation.

2.5. Performance measurement

The performance measurement of predictive models is crucial for their practical application. A common measure regarding predictive models, including neural networks, is the use of square average error. Many times prediction errors are too large in the context of real applications because of the stochastic components present in the financial series. For this reason it is necessary to use additional performance measures to test the validity of the predictive models. Examples of additional performance measures are: the measure of generated profit, the measure of correct prediction of different value, the measure of correlation between the real values and the predicted values, the measure of error towards the real value of predicted value and other similar measures [20].

In conclusion, we can say that neural networks are efficient tools to detect nonlinear relations that rule, at least partially, the behavior of time series. As the majority of financial data series contain nonlinear components, neural networks are good candidates to predict this type of series. Because financial data series have important stochastic components, is very important to build minimal models, that estimate sufficiently well nonlinear relation, and in the same time, it doesn't incorporate the stochastic noise.

3. PRACTICAL IMPLEMENTATION OF THE NEURAL NETWORK

The practical part of this paper is related to the implementation of a neural network that offers the possibility to analyze and predict data series. In this simulation we have tried to approximate and to predict some financial data series.

The parameters that are influences the learning process are: the training data set, the number of epochs (number of the presentation of the training data set), learning rate, the activation functions for the neurons contained in the hidden layer, the number of neurons in the hidden layer.

We have built a practical application, based on a MLP neural network, in order to test the prediction capabilities of neural networks [4].

The architecture of the neural network used in our simulation is corresponding to a MLP neural network with one hidden layer [11]:

- The input layer contains n input neurons, *n* representing the dimensionality of the input space $x_i = \left(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}\right) \in \mathbf{R}^n$. The bias can be considered explicitly or implicitly.
- The hidden layer having a number of hidden neurons equal to the dimension of the training set $T = \left\{\left(\boldsymbol{x}_i, f\left(\boldsymbol{x}_i\right)\right) \middle| i = 1,2,\ldots,N\right\}$. The activation functions of the hidden neurons are Green functions $G\left(\boldsymbol{x} - \boldsymbol{x}_k\right)$ [11]. The dimension of the hidden layer can be reduced using an unsupervised clustering algorithm;
- The output layer contains one single output layer having as activation function a linear function or a special weighted functions of the output values generated by the neurons in the hidden layer [6];

Synaptic weights:

- The weights between the input layer and the hidden layer are included in the form of the activation functions of the hidden neurons. The vector $\boldsymbol{w} = \left(w_1, w_2, \ldots, w_N\right)$ represents the weights between the hidden layer and the output layer.

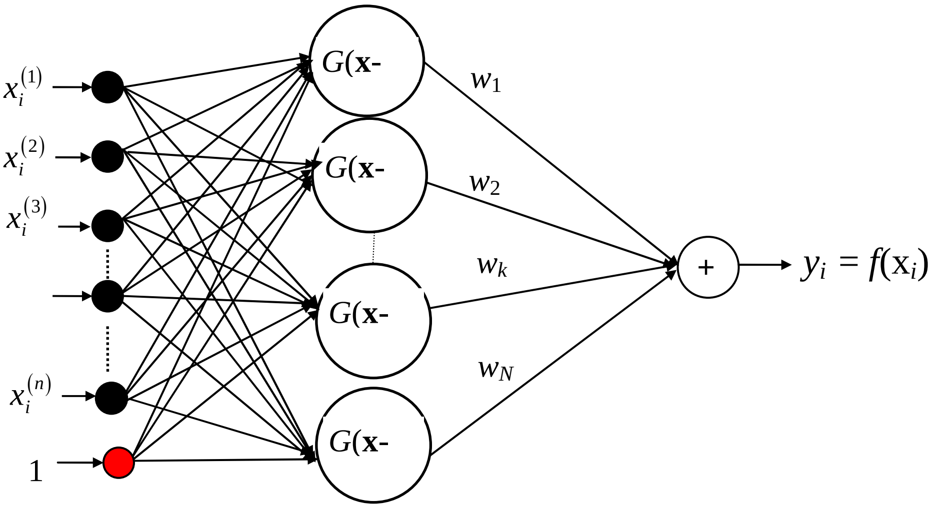

Fig. 1: **Architecture of the neural network used for simulations.**

The neural network was trained using an original learning algorithm, based on the back-propagation learning strategy. For learning we have used a training set containing financial data series, like the exchange rate between RON and EUR or RON and USD. The learning set, corresponds to the time frame January 2004- June 2005, and was obtained from the official Web-site of the National Bank of Romania http://www.bnr.ro (*Banca Nationala a Romaniei*). After learning we have performed a testing phase, in order to measure the accuracy of the trained neural network.

Number of epochs	Learning error
10	0.0289682211941586
50	0.0135544478100808
100	0.00669238729756063
500	0.00268232311020435
1000	0.00192883390014049
5000	0.00163946058914474
10000	0.00157560371972362

Table 1.: Results of the learning phase: the learning error obtained for different number of epochs.

4. CONCLUSIONS

The paper presents a general model of the application of neural networks in the domain of modeling and prediction of economic and financial time series. The synthesis is based on two theoretical blocks, namely, the approximation properties of the neural networks and the regularization theory. The results presented in the paper confirm the fundamental role of the theoretical models for the application of neural networks for prediction time series.

REFERENCES:

[1]. Adlar, J. Kim, Christian, R. Shelton (2002) Modeling Stock Order Flows and Learning Market-Marking from data, AIM-2002-009, MIT.
[2]. András, P. (2000) Rețele neuronale pentru aproximare şi predicția seriilor de timp, Universitatea „Babeş-Bolyai" Cluj Napoca.
[3]. Enăchescu, C. (1998) Bazele teoretice ale rețelelor neuronale, Editura Casa Cărţii de Ştiinţă Cluj-Napoca, (in Romanian).
[4]. Enăchescu, C. (1997) Elemente de inteligenţă artificială. Calcul neuronal., Universitatea „Petru Maior" Târgu-Mureş, 1997, (in Romanian).
[5] Enăchescu, C. (1995) Properties of Neural Networks Learning, 5th International Symposium on Automatic Control and Computer Science, SACCS'95, Vol.2, 273-278, Technical University "Gh. Asachi" of Iasi, Romania.
[6] Enăchescu, C. (1996) Neural Networks as approximation methods. International Conference on Approximation and Optimisation Methods, ICAOR'96, " Babes-Bolyai University ", Cluj-Napoca.
[7] Enăchescu, C., (1995) Learning the Neural Networks from the Approximation Theory Perspective. Intelligent Computer Communication ICC'95 Proceedings, 184-187, Technical University of Cluj-Napoca, Romania.
[8]. Haykin, S. (1997), *Neural Networks. A Comprehensive Foundation*. IEEE Press, MacMillian, 1994.
[9] Hecht-Nielsen, R. (1989). Theory of the back propagation Neural Network. IEEE International Joint Conference on Neural Networks, 1, 593-605.
[10]. Hertz, J., Krocgh, A., Palmers, R. (1997), *Introduction to the theory of Neural Networks*, Edison Wesley Publishing, 1992.

[11] Girosi, F., T. Pogio (1990). Networks and the Best Approximation Property. Biological Cybernetics, 63, 169-176.
[12] Haykin, S. (1994), Neural Networks. A Comprehensive Foundation. IEEE Press, MacMillian.
[13] Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer Feedforward Networks Are Universal Approximators. Neural Networks, 2, 359-366.
[14]. Lackes, R. and Mack D.: Neural Networks. Basics and Applications, Springer-Verlag Berlin Heidelberg, 1998.
[15] Pogio, T., F. Girosi (1990). Networks for Approximation and Learning. Proceedings of the IEEE, Vol. 78, 9, 1481-1497.
[16] Petri, A. J. (1991). A Nonlinear Network Model for continuous learning. Neurocomputing 3, 157-176.
[17] Tichonov, A.N., Arsenin, V.A. (1997), *Solutions of Ill-posed Problems*. Washington, DC: W.H. Winston, 1977.
[18] Tikhonov, A.N. (1997), *Solution of incorrectly formulated problems and regularization method*. Soviet Math. Dokl., Vol. 4, 1035-1038, 1963.
[19] White, D.A., (1989) Learning in artificial neural networks: A statistical perspective. Neural Computation 1, 425-464.
[20]. Weigend A.: Time series prediction, Addison-Wesley, 1994.